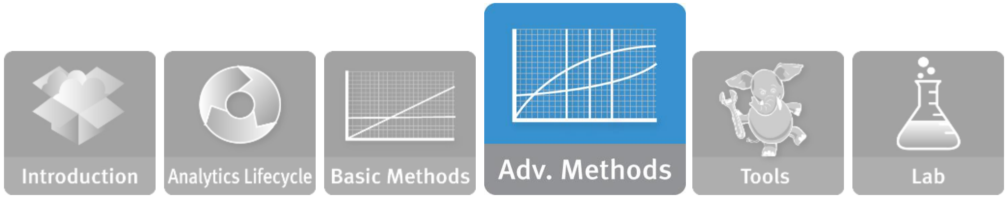
Introduction

Big Data Analytics

Presented by: Dr Sherin El Gokhy

Adv. Methods

Module 4 – Advanced Analytics - Theory and Methods



Module 4: Advanced Analytics – Theory and Methods

Part 6: Decision Trees

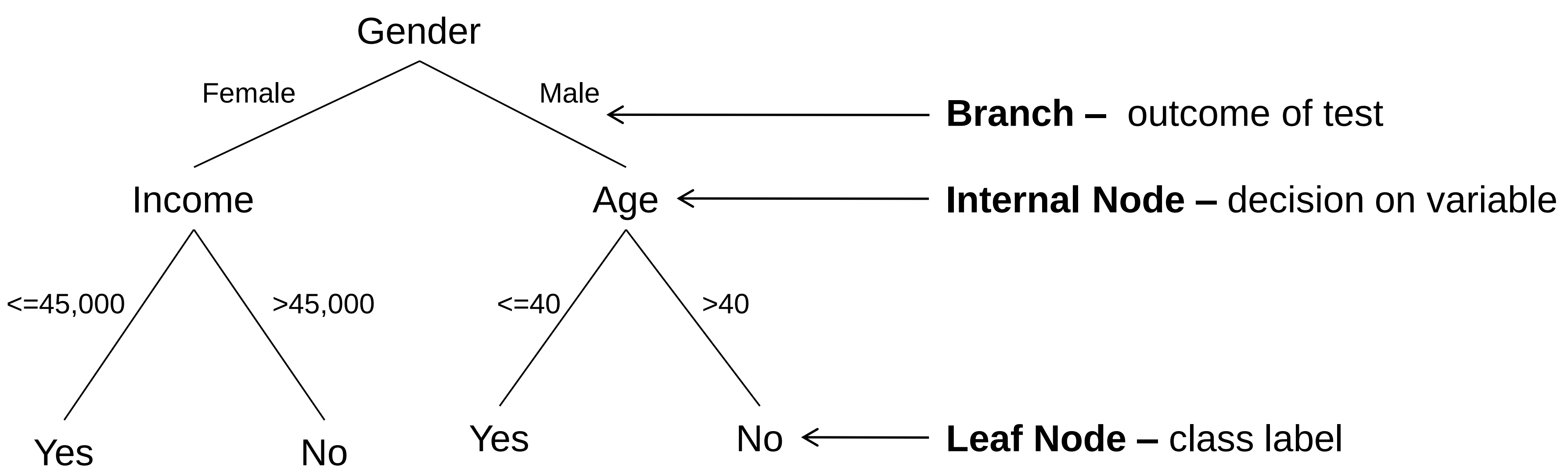
During this Part the following topics are covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

Decision Tree Classifier - What is it?

- Used for classification:
  - Returns probability scores of class membership
    - Well-standardized, like logistic regression
    - Assigns label based on highest scoring class
    - Some Decision Tree algorithms return simply the most likely class
  - Regression Trees: a variation for regression
    - Returns average value at every node
    - Predictions can be discontinuous at the decision boundaries
- Input variables can be continuous or discrete
- Output:
  - A tree that describes the decision flow.
  - Leaf nodes return either a probability score, or simply a classification.
  - Trees can be converted to a set of "decision rules“
    - "IF income < $50,000 AND mortgage_amt > $100K THEN default=T with 75% probability“

Decision Tree – Example of Visual Structure



Branches refer to the outcome of a decision.

When the decision is numerical, the "greater than" branch is usually shown on the right and "less than" on the left. Depending on the nature of the variable, you may need to include an "equal to" component on one branch.

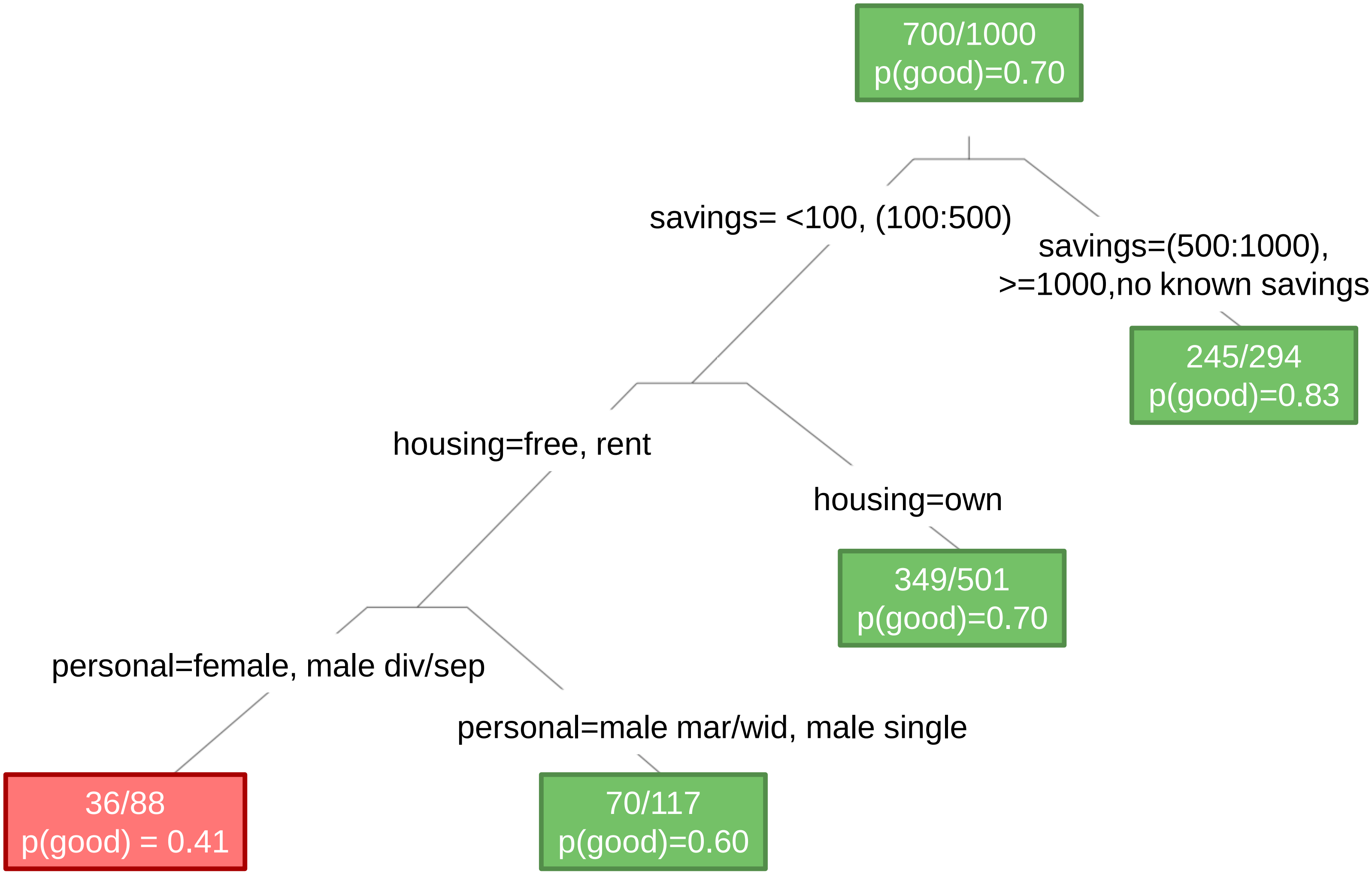
Internal Nodes are the decision or test points. Each refers to a single variable or attribute.

Note that you do not make the decision using all the attributes in each case

Decision Tree Classifier - Use Cases

- When **a series of questions (yes/no) are answered** to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor’s evaluation of a patient
- When “if-then” conditions are preferred to linear models.
  - Customer segmentation to predict response rates to marketing and promotions
  - Financial decisions such as loan approval…. computers can use the logical “if-then” statements to predict whether the customer will default on the loan
  - Fraud detection
- Short Decision Trees (where we have limited the number of splits) are often used as components (called "weak learners" or "base learners") in ensemble techniques (a set of predictive models which will all vote and we take decisions based on the combination of the votes) such as Random forests

Example: The Credit Prediction Problem



General Algorithm

- To construct tree T from training set S
  - If all examples in S belong to some class in C, or S is sufficiently "pure", then make a leaf labeled C.
  - Otherwise:
    - select the “most informative” attribute A
    - partition S according to A’s values
    - recursively construct sub-trees T1, T2, ..., for the subsets of S
- There are several algorithms that implement Decision Trees and the methods of tree construction vary with each one of them. CART,ID3 and C4.5 are some of the popular algorithms.

Step 1: Pick the Most “Informative" Attribute

- Entropy-based methods are one common way

$$H = -\sum_{c} p(c) \log_2 p(c)$$

- H = 0 if p(c) = 0 or 1 for any class
  - So for binary classification, H=0 is a "pure" node
- H is maximum when all classes are equally probable
  - For binary classification, H=1 when classes are 50/50

Step 1: Pick the most "informative" attribute (Continued)

- First, we need to get the base entropy of the data (Unconditional entropy)

$$\boxed{\begin{aligned} H_{credit} &= -(0.7\log_2(0.7) + 0.3\log_2(0.3)) \\ &= 0.88 \end{aligned}}$$

Step 1: Pick the Most "Informative" Attribute (Continued)
Conditional Entropy

$$H_{attr} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

- **The weighted sum of the class entropies for each value of the attribute**
- In English: attribute values (home owner vs. renter) give more information about class membership
  - "Home owners are more likely to have good credit than renters"
  - So the attribute value Housing will give more information about the class membership for credit good.
- **Conditional entropy should be lower than unconditioned entropy**

Conditional Entropy Example

	for free	own	rent
P(housing)	0.108	0.713	0.179
P(bad | housing)	0.407	0.261	0.391
p(good | housing)	0.592	0.739	0.601

$$\begin{aligned} H_{(housing|credit)} &= -[0.108 * (0.407 \log_2(0.407) + 0.592 \log_2(0.592)) \\ &\quad + 0.713 * (0.261 \log_2(0.261) + 0.739 \log_2(0.739)) \\ &\quad + 0.179 * (0.391 \log_2(0.391) + 0.601 \log_2(0.601))] \\ &= 0.868 \end{aligned}$$

Step 1: Pick the Most “Informative" Attribute (Continued) Information Gain

$$\text{InfoGain}_{attr} = H - Hattr$$

- The information that you gain, by knowing the value of an attribute
- So the "most informative" attribute is the attribute with the highest InfoGain

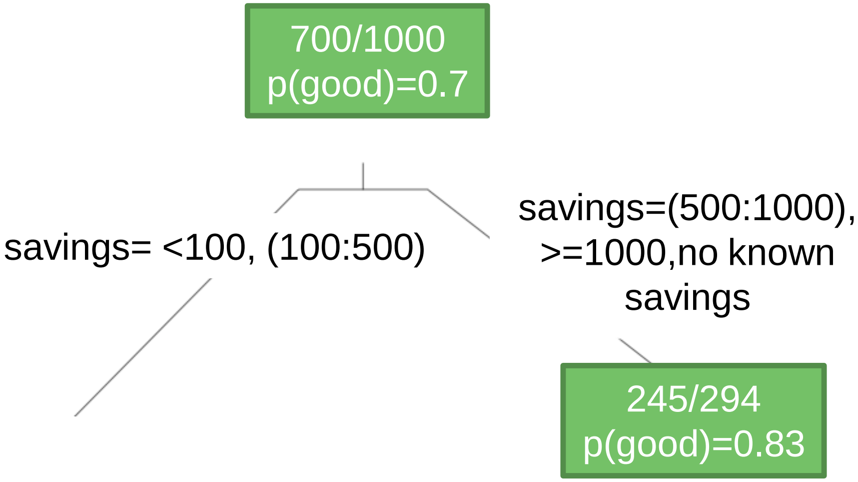
Back to the Credit Prediction Example

$$\begin{aligned}\text{InfoGain}_{credit} &= H_{credit} - H_{housing|credit} \\ &= 0.88 - 0.86 \\ &\approx 0.013\end{aligned}$$

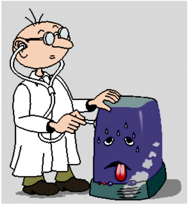
Attribute	InfoGain
job	0.001
housing	0.013
personal_status	0.006
savings_status	***0.028***

Step 2 & 3: Partition on the Selected Variable

- Step 2: Find the partition with the highest InfoGain
  - In our example the selected partition has InfoGain = 0.028
- Step 3: At each resulting node, repeat Steps 1 and 2
- Step 1: Pick the Most “Informative" Attribute
  - until node is "pure enough"
- Pure nodes => no information gain by splitting on other attributes



Diagnostics



- Hold-out data
- ROC/AUC
- Confusion Matrix
- FPR/FNR, Precision/Recall
- Do the splits (or the "rules") make sense?
  - What does the domain expert say?
- How deep is the tree?
  - Too many layers are prone to over-fit
- Do you get nodes with very few members?
  - Over-fit

Decision Tree Classifier - Reasons to Choose (+) & Cautions (-)

Reasons to Choose (+)	Cautions (-)
Takes any input type (numeric, categorical) In principle, can handle categorical variables with many distinct values (ZIP code)	Decision surfaces can only be axis-aligned
Robust with redundant variables, correlated variables	Tree structure is sensitive to small changes in the training data
Naturally handles variable interaction	A "deep" tree is probably over-fit Because each split reduces the training data for subsequent splits
Handles variables that have non-linear effect on outcome	Not good for outcomes that are dependent on many variables Related to over-fit problem, above
Computationally efficient to build	Doesn't naturally handle missing values; However most implementations include a method for dealing with this
Easy to score data	In practice, decision rules can be fairly complex
Many algorithms can return a measure of variable importance	
In principle, decision rules are easy to understand	

Which Classifier Should I Try?

Typical Questions	Recommended Method
Do I want class probabilities, rather than just class labels?	Logistic regression Decision Tree
Do I want insight into how the variables affect the model?	Logistic regression Decision Tree
Is the problem high-dimensional?	Naïve Bayes
Do I suspect some of the inputs are correlated?	Decision Tree Logistic Regression
Do I suspect some of the inputs are irrelevant?	Decision Tree Naïve Bayes
Are there categorical variables with a large number of levels?	Naïve Bayes Decision Tree
Are there mixed variable types?	Decision Tree Logistic Regression
Is there non-linear data or discontinuities in the inputs that will affect the outputs?	Decision Tree

Check Your Knowledge

Your Thoughts?

1. How do you define information gain?
2. For what conditions is the value of entropy at a maximum and when is it at a minimum?
3. List three use cases of Decision Trees.
4. What are weak learners and how are they used in ensemble methods?
5. Why do we end up with an over fitted model with deep trees and in data sets when we have outcomes that are dependent on many variables?
6. What classification method would you recommend for the following cases:
   - High dimensional data
   - Data in which outputs are affected by non-linearity and discontinuity in the inputs

Module 4: Advanced Analytics – Theory and Methods

Part 6: Decision Trees - Summary

During this Part the following topics were covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

Lab Exercise 9: Decision Trees

This lab is designed to investigate and practice Decision Tree models covered in the course work.

After completing the tasks in this lab you should be able to:

- Use R functions for Decision Tree models
- Predict the outcome of an attribute based on the model

Lab Exercise 9: Decision Trees - Workflow

1
- Set the Working Directory

2
- Read in the Data

3
- Build the Decision Tree

4
- Plot the Decision Tree

5
- Prepare Data to Test the Fitted Model

6
- Predict a Decision from the Fitted Model

Thanks